

COP 3330 Suggested Exercises for Week 11

TreeSet

1) Play a game of The Price is Right. Ask the user to enter the correct price of an item, and then after that, get guesses for the price of the item. After each guess, report the largest guess that is less than or equal to the correct price of the item. (Note: You don't really need a TreeSet to do this, but it's a nice simple exercise to practice using a TreeSet.)

2) Write a program that allows you to (a) add a person to a list, (b) remove a person from a list, (c) find the first person alphabetically on the list, (d) find the last person alphabetically on a list, (e) search for a person on the list, (f) find the next person alphabetically on the list after a particular person, (g) find the previous person alphabetically on the list before a particular person.

3) A classic problem that can be solved by a TreeSet is the "rolling median" problem, which goes as follows:

Given a list of **distinct** integers, one by one added to a set of data, after each integer is added, report the median of the data. The median of an odd number of values is the unique middle value. The median of an even number of values is the average of the two middle values.

Note: There's a nice trick to this one. If you are getting stuck, feel free to come to the instructor to get the hint.

TreeMap

4) Write a program that reads in a list of n votes for Mayor (each vote is a string), and then prints out a list, sorted in alphabetical order of the number of votes each person has received.

5) Imagine being given a long spaceway line as stops are added over time. Let the line start at millimeter marker 0 and end at millimeter marker 10^{12} . Over time, stops are added at different millimeter markers. When a stop is added, a name is given to the stop. For example, at mm marker 123456789 we add the stop, "Saturn_Safestop". Write a program that reads in stops that are added and queries for the nearest stop to a given mile marker. The queries should be answered with the name of the stop and how many millimeters away it is from the current stop.

6) Write a program that reads in new countries as they are created along with their capitals. Whenever the user wants a full print out of the list of countries and their capitals, sorted by the countries in alphabetical order, provide that list.

7) This one's a bit challenging, but it can be solved with a nice use of a TreeMap. Try this problem from Kattis:

<https://open.kattis.com/problems/bst>

PriorityQueue

8) Use a PriorityQueue and other built in data structures to solve this Kattis problem:

<https://open.kattis.com/problems/guessthedatastructure>

9) Use a Priority Queue to solve this Kattis problem:

<https://open.kattis.com/problems/janeeyre>

10) This is a bit advanced, but read about Prim's Algorithm to solve the minimum spanning tree problem and use Java's built in PriorityQueue to implement the algorithm. The algorithm takes in information about connections between vertices (you can think of these as roads with particular costs to construct between two cities) and determines the minimum cost to build a road network that connects all of the cities.